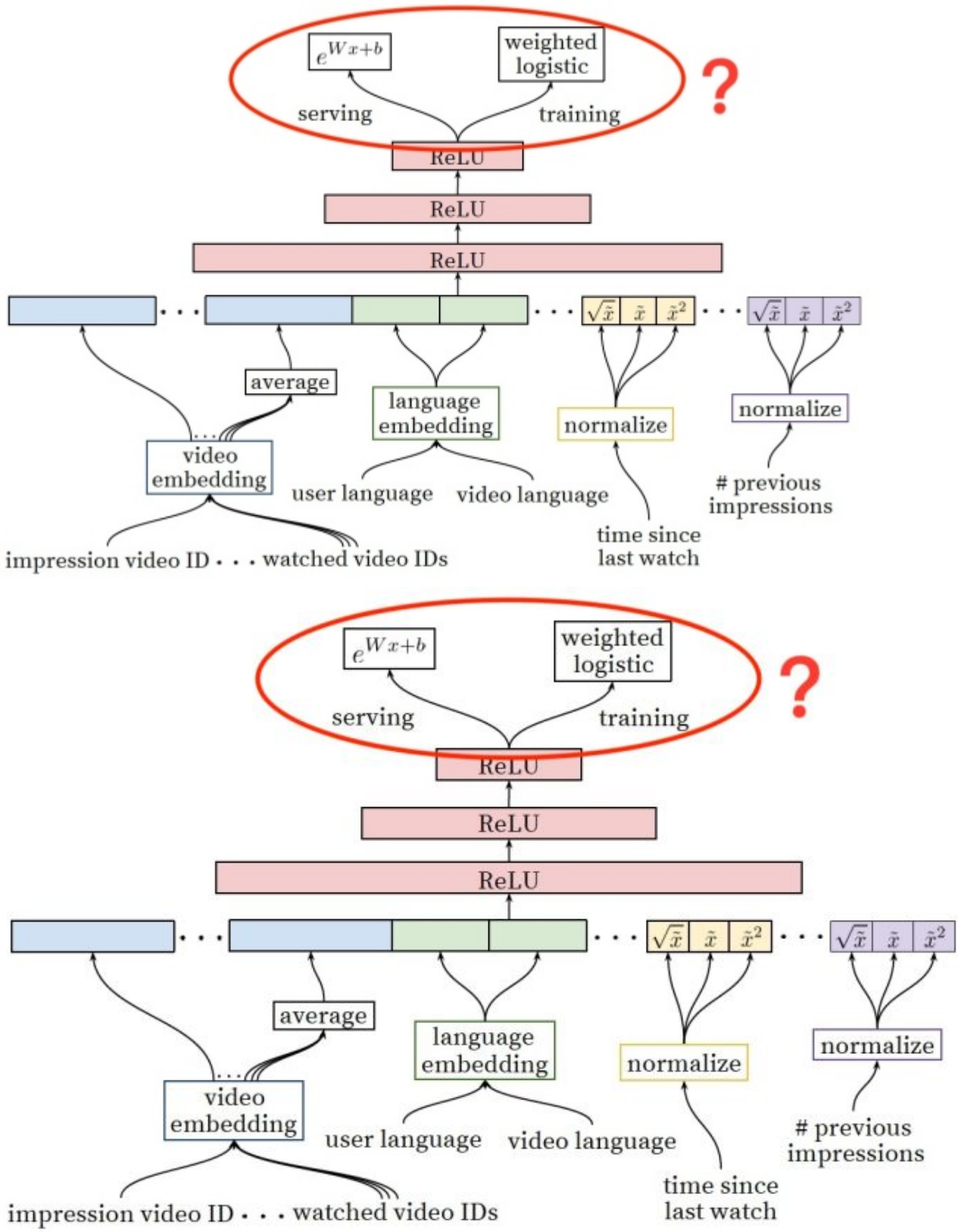
这里是「**王喆的机器学习笔记**」的第十二篇文章，今天我们着手来彻底解决一个许久以来悬而未决的问题，也至少有十几位专栏读者通过留言和私信的方式询问我这个问题，这个问题就是**YouTube深度学习推荐系统中模型serving的问题**。

不了解YouTube深度学习推荐系统的同学可以回顾一下我之前的两篇专栏文章，以及YouTube的论文原文：

1. 王喆：重读Youtube深度学习推荐系统论文，字字珠玑，惊为神文
2. 王喆：YouTube深度学习推荐系统的十大工程问题
3. [Youtube] Deep Neural Networks for YouTube Recommendations (Youtube 2016)

这里我们再详细陈述一下这个问题：



YouTube深度学习推荐系统中Ranking Model的架构图

上图是YouTube推荐系统排序模型（Ranking Model）的架构图，我们不再重复讲解模型的细节，而是把关注的焦点放在最后的输出层：

为什么**Ranking Model**采用了**weighted logistic regression**作为输出层？在模型**serving**过程中又为何没有采用**sigmoid**函数预测正样本的**probability**，而是使用 e^{Wx+b} 这一指数形式预测用户观看时长？

对于传统的深度学习架构，输出层往往采用LR或者Softmax，在线上预测过程中，也是原封不动的照搬LR或者softmax的经典形式来计算点击率（广义地说，应该是正样本概率）。

而YouTube这一模型的神奇之处在于，输出层没有使用LR，而是采用了Weighted LR，模型serving没有采用sigmoid函数的形式，而是使用了

$$e^{Wx+b}$$

这一指数形式。按照原文说法，这样做预测的就是用户观看时长？？没有任何其他表情能像这位小哥一样表达我初读论文的感受。。What？？？

搞清楚这件事情并不是一件容易的事情，我们要从逻辑回归的本质意义上开始。

几乎所有算法工程师的第一堂课就是逻辑回归，也肯定知道逻辑回归的数学形式就是一个线性回归套sigmoid函数：

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

逻辑回归的数学形式

但为什么选择sigmoid函数？难道仅仅是sigmoid函数能把值域映射到0-1之间，符合概率的物理意义这么简单吗？

答案显然不会这么肤浅。

为解释这个问题，首先我们需要定义一个新的变量——**Odds**，中文可以叫发生比或者机会比。

$$Odds = \frac{p}{1-p}$$

$$Odds = \frac{p}{1-p}$$

Odds的定义

假设一件事情发生的概率是p，那么**Odds就是一件事情发生和不发生的比值**。

如果对Odds取自然对数，再让ln(Odds)等于一个线性回归函数，那么就得到了下面的等式。

$$logit(p) = \ln(\frac{p}{1-p}) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

$$logit(p) = \ln(\frac{p}{1-p}) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

其中ln(p/(1-p))就是大名鼎鼎的**logit函数**，logistics regression又名logit regression，上面的式子就是逻辑回归的由来。我们再做进一步运算，就可以转变成我们熟悉的逻辑回归的形式：

$$ln(\frac{p}{1-p}) = \theta^T x \Rightarrow \frac{p}{1-p} = e^{\theta^T x} \Rightarrow p = \frac{1}{1+e^{-\theta^T x}} \Rightarrow p = sigmoid(\theta^T x)$$

到这里大家应该已经完全明白了LR的推导过程了。

那么再对

$$ln(Odds) = \theta^T x$$

这个等式做一个小小的转换，两边取自然底数：

$$ln(Odds) = \theta^T x \Rightarrow Odds = e^{\theta^T x} = YouTubeServingFunction$$

大家看到了吗，Youtube的Serving函数

$$e^{Wx+b}$$

计算的不是别的，正是Odds！

但我们还没有到达终点，因为Youtube要预测的明明是用户观看时长，怎么就成了Odds了？

这就要提到YouTube采用的独特的训练方式Weighted LR，这里的Weight，对于正样本i来说就是观看时长Ti，对于负样本来说，则指定了单位权重1。

Weighted LR的特点是，正样本权重w的加入会让正样本发生的几率变成原来的w倍，也就是说样本i的Odds变成了下面的式子：

$$Odds(i) = \frac{w_i p}{1 - w_i p}$$

由于在视频推荐场景中，用户打开一个视频的概率p往往是一个很小的值，因此上式可以继续简化：

$$Odds(i) = \frac{w_i p}{1 - w_i p} \approx w_i p = T_i p = E(T_i)$$

而且由于YouTube采用了用户观看时长Ti作为权重，因此式子进一步等于Ti*p，这里真相就大白了，由于p就是用户打开视频的概率，Ti是观看时长，因此Ti*p就是用户观看某视频的期望时长！

因此，YouTube采用

$$e^{Wx+b}$$

这一指数形式预测的就是曝光这个视频时，用户观看这个视频的时长的期望！利用该指标排序后再进行推荐，是完全符合YouTube的推荐场景和以观看时长为优化目标的设定的。

再简要总结一下YouTube Ranking Model的Serving过程要点。

1. e^{Wx+b} 这一指数形式计算的是Weighted LR的Odds；
2. Weighted LR使用用户观看时长作为权重，使得对应的Odds表示的就是用户观看时长的期望；
3. 因此，Model Serving过程中e^{Wx+b} 计算的正是观看时长的期望。

最后按惯例给大家留一个讨论的问题，欢迎大家各抒己见：

训练Weighted LR一般来说有两种办法：

1. 将正样本按照weight做重复sampling，然后输入模型进行训练；
2. 在训练的梯度下降过程中，通过改变梯度的weight来得到Weighted LR。

问题是这两种训练方法得到的结果有没有不同？有没有其他Weighted LR的训练方法？

希望这篇文章能够终结大家对于YouTube模型Serving问题的疑惑，如果有不明白的同学，欢迎大家在知乎专栏或者我的微信公众号“王喆的机器学习笔记”（**wangzhenotes**）留言讨论，想进一步交流的同学也可以通过我的公众号加我的个人微信一同探讨，谢谢。

> 本文转载自 揭开YouTube深度推荐系统模型Serving之谜 - 王喆的文章 - 知乎 https://zhuanlan.zhihu.com/p/61827629